**一、这两个文件的区别与作用**

| **文件名** | **含义** | **数据来源** | **推荐系统用途** |
| --- | --- | --- | --- |
| CRISPR\_gene\_effect.csv | 细胞系中某基因被 KO 后的生长抑制程度（负值 = 更依赖） | Achilles (Avana) CRISPR KO | 🧬 crispr\_score, crispr\_consistency |
| CRISPR\_gene\_dependency.csv | 某基因对细胞的依赖性概率（0~1之间） | 基于 Bayes 模型从 KO 数据估算 | ✅ 可构建 dependency\_score |

**✅ 二、这些数据在推荐系统中的典型特征字段**

| **字段名** | **解释** | **构建方法** |
| --- | --- | --- |
| crispr\_mean\_effect | KO 后平均影响值（G12C 阳性样本中） | 平均 gene effect |
| crispr\_consistency | 多细胞系标准差的倒数 | 1 - std |
| crispr\_dependency\_score | 基因在 G12C 阳性细胞中的依赖概率 | 平均 dependency 值 |
| crispr\_effect\_LFC | KO 后生长情况 vs 非G12C 对照 | 差异分析 |

**✅ 三、你现在可以做的事**

**1️⃣ 已准备文件：**

* CRISPR\_gene\_effect.csv
* CRISPR\_gene\_dependency.csv
* sample\_info.csv（用于筛 G12C 样本）

**2️⃣ 可以立刻执行的分析目标：**

| **任务** | **输出字段** | **用于推荐系统字段** |
| --- | --- | --- |
| 筛选 KRAS G12C 细胞系 | sample\_info.csv + mutation.csv | 生成目标 DepMap\_ID 列表 |
| 计算 KO 分值均值与一致性 | CRISPR\_gene\_effect.csv | crispr\_mean\_effect, crispr\_consistency |
| 计算依赖性评分 | CRISPR\_gene\_dependency.csv | crispr\_dependency\_score |
| 组间差异分析（G12C+ vs -） | KO effect, dependency | crispr\_effect\_LFC |

**✅ 我可以现在为你做什么？**

我可以立刻：

1. ✅ 写出自动整合这两个文件的脚本
2. ✅ 输出 crispr\_mean, crispr\_consistency, crispr\_dependency\_score 等字段
3. ✅ 与 RNAseq 特征合并，进一步构建推荐系统输入矩阵

**分析主策略：以 KRAS G12C 细胞系为主导来筛样本**

这个策略是指：

在分析 CRISPR 数据时，我将“KRAS G12C 阳性”的细胞系作为一个**目标组（case group）**，  
与其余非 G12C 细胞系作为\*\*对照组（control group）\*\*进行比较或统计。

**✅ 这样做的意义是：**

| **目的** | **举例** | **推荐系统价值** |
| --- | --- | --- |
| 📊 计算 KO 或依赖性分数 | **DUSP6 在 G12C 细胞中平均 KO 得分是多少？** | 构建 crispr\_mean\_effect 字段 |
| 📉 计算 KO 得分变异性 | **KRAS G12C 细胞中 KO 某基因的分数稳定吗？** | 构建 crispr\_consistency |
| 📈 差异性评估 | **AURKA 在 G12C vs 非 G12C 中 KO 效应差异大吗？** | 构建 crispr\_effect\_LFC、crispr\_effect\_pval |

这与 RNAseq 差异分析中 log2FoldChange (G12C vs non) 是一模一样的逻辑，只不过把表达量换成了 **KO 效应值** 或 **依赖性分数**。

**✅ 所以我们会按如下样本分组处理：**

| **分组** | **来源** | **用途** |
| --- | --- | --- |
| ✅ G12C 阳性细胞系 | mutation\_calls.csv 中 Protein\_Change 包含 G12C 的 DepMap\_ID | 主分析组 |
| 🟦 G12C 阴性细胞系（NSCLC背景） | 其他 NSCLC 细胞系 | 对照组，用于差异分析 |
| 🟥 可选：所有细胞系 | 如果想算 global mean | 构建全局 crispr\_score\_mean |

📦 总结：

* 我们会“以 KRAS G12C 阳性细胞系为核心”来提取：
  + KO 平均效应
  + 一致性（方差）
  + 依赖性分数
  + 与非G12C样本的差异 LFC

**extract\_crispr\_features\_kras\_g12c.py**

**脚本功能概览：**

1. 📥 读取：
   * CRISPR\_gene\_effect.csv
   * CRISPR\_gene\_dependency.csv
   * CCLE\_mutations.csv
   * sample\_info.csv
2. 🔎 自动识别 KRAS G12C 阳性细胞系
3. 🧬 对每个基因计算：
   * crispr\_mean\_effect（KO后在G12C细胞系中的平均效应）
   * crispr\_consistency = 1 - std（KO一致性）
   * crispr\_effect\_LFC 与 pval（与非G12C对照比较）
   * crispr\_dependency\_score（依赖性分数）
4. 📤 输出表格：

复制编辑

CRISPR\_features\_KRAS\_G12C.csv

**📊 输出字段结构：**

| **gene** | **crispr\_mean\_effect** | **crispr\_std\_effect** | **crispr\_consistency** | **crispr\_effect\_LFC** | **crispr\_effect\_pval** |
| --- | --- | --- | --- | --- | --- |

脚本内关于从突变数据表内寻找标记为KRAS G12C突变的细胞系去CRISPR数据表内对比标记的功能部分的代码的解释

**✅ 代码原文**

python

复制编辑

# ==== 获取 KRAS G12C 阳性 DepMap ID ====

g12c\_samples = mut\_df[

(mut\_df["Hugo\_Symbol"] == "KRAS") &

(mut\_df["Protein\_Change"].str.contains("G12C", na=False))

]["DepMap\_ID"].unique().tolist()

print(f"✅ 找到 KRAS G12C 阳性细胞系数：{len(g12c\_samples)}")

**✅ 背景说明**

CCLE\_mutations.csv 文件是一个 **“突变注释表”**，每一行代表一个细胞系中发现的一个突变。

字段包括（示意）：

| **DepMap\_ID** | **Hugo\_Symbol** | **Variant\_Classification** | **Protein\_Change** |
| --- | --- | --- | --- |
| ACH-000001 | TP53 | Missense\_Mutation | p.R273H |
| ACH-000002 | **KRAS** | Missense\_Mutation | **p.G12C** |
| ACH-000003 | KRAS | Missense\_Mutation | p.G12D |

**✅ 每一行代码解析**

**1️⃣ g12c\_samples = mut\_df[...]**

创建一个新变量 g12c\_samples，用于存储符合筛选条件的细胞系 ID 列表  
mut\_df[...] 是在突变数据中做筛选操作

**2️⃣ mut\_df["Hugo\_Symbol"] == "KRAS"**

找出 Hugo\_Symbol 字段（即基因名）为 "KRAS" 的突变行  
🚨 注意：此处是筛选突变发生在 **KRAS 基因** 的行

**3️⃣ mut\_df["Protein\_Change"].str.contains("G12C", na=False)**

找出 Protein\_Change 字段中包含 G12C 字样的行

* .str.contains("G12C")：查找包含“G12C”的蛋白质变异，如 p.G12C
* na=False：忽略空值，不报错

👉 **最终目的是筛出突变为 KRAS G12C 的所有记录**

**4️⃣ &**

逻辑“与”：同时满足 “基因是 KRAS” **并且** “变异是 G12C”

**5️⃣ ["DepMap\_ID"]**

只取这些行的 DepMap\_ID 字段，用于定位细胞系

**6️⃣ .unique()**

去重：同一个细胞系可能出现多次变异，只保留唯一 ID

**7️⃣ .tolist()**

将结果变成 Python 的列表（List[str]）

最终结果是：

python

复制编辑

g12c\_samples = ["ACH-000002", "ACH-001234", ...]

**8️⃣ print(f"...")**

打印找到的 KRAS G12C 阳性细胞系数量

比如输出：

text

复制编辑

✅ 找到 KRAS G12C 阳性细胞系数：27

**✅ 整体功能总结**

🔍 **目的**：  
从突变注释表中筛选出所有 **发生了 KRAS G12C 突变的细胞系 ID**

⚙️ **用于后续的功能组学分析**，例如：

* 表达对比（G12C vs 非G12C）
* CRISPR KO 对比（是否在 G12C 样本中敲除后效果不同）
* 构建推荐系统的 “目标样本组”

🧠 你可以理解为：\*\*这是定义推荐系统的“病例组”\*\*的第一步！